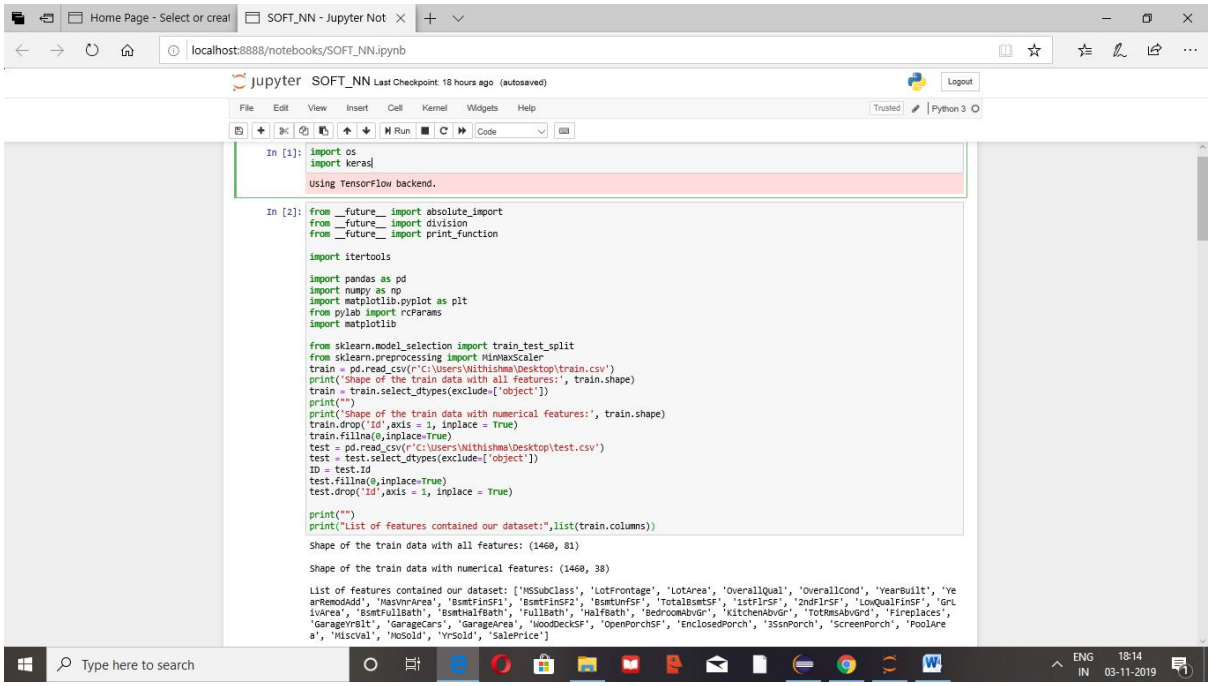
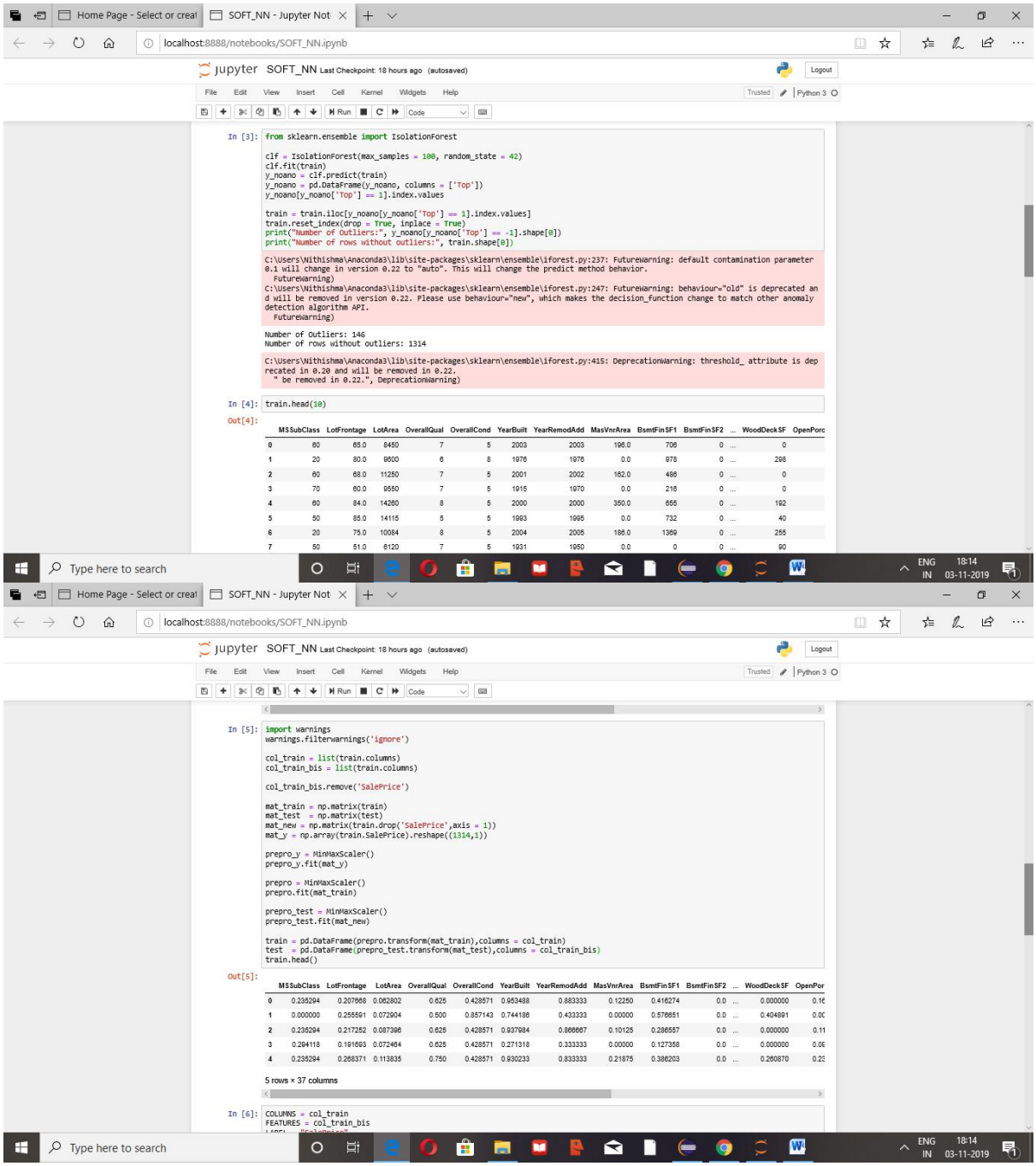
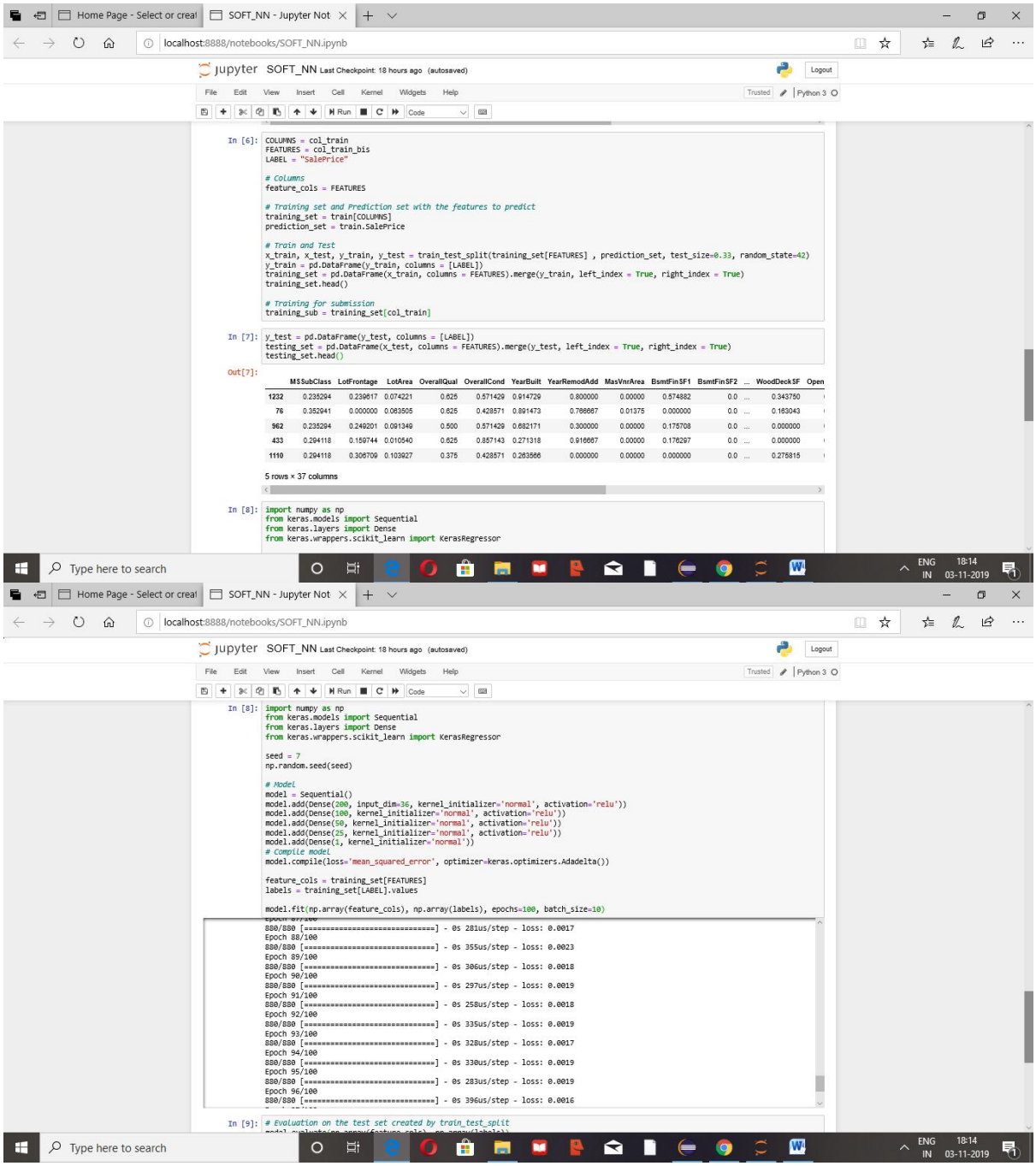
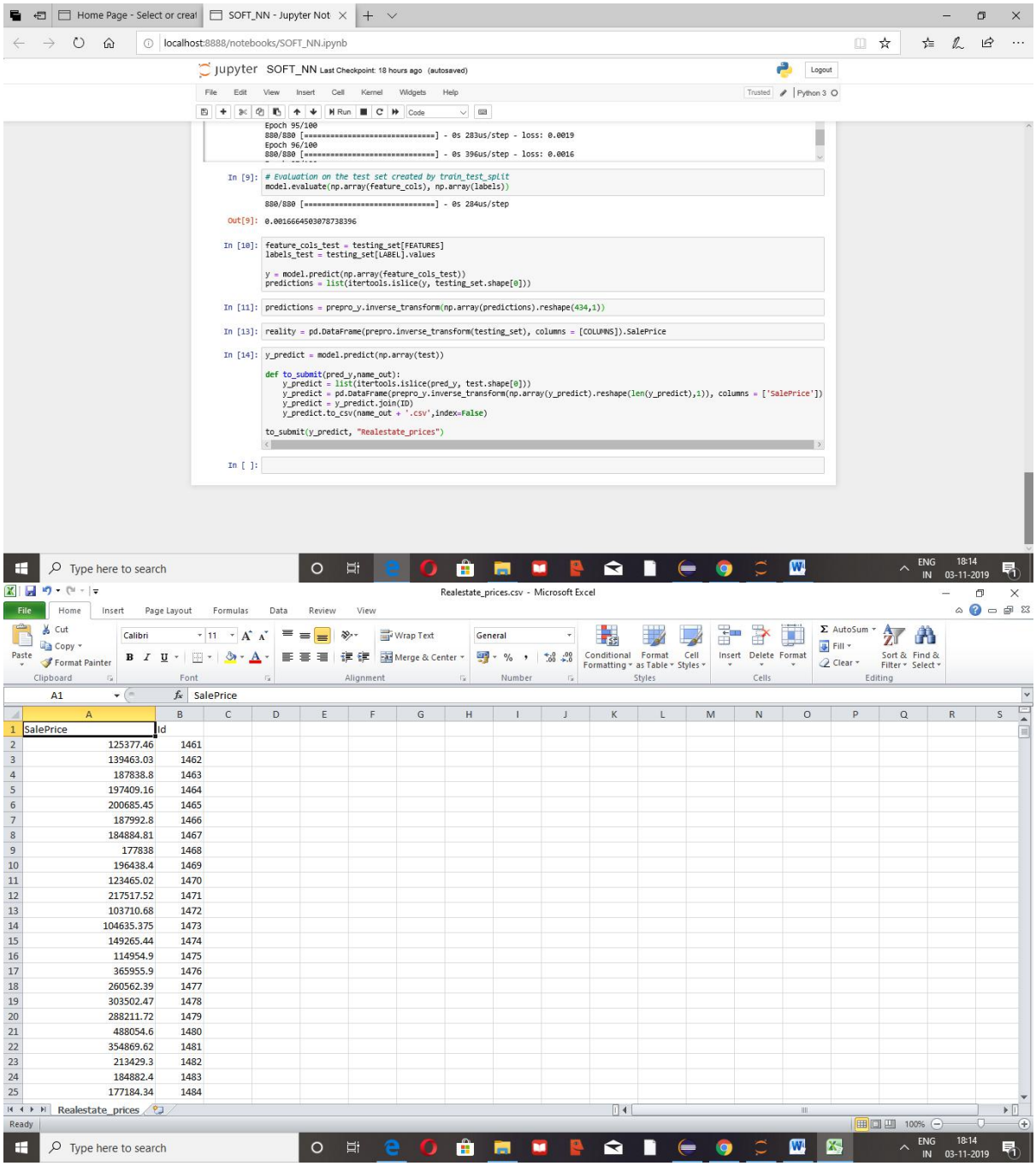

Home Page - Select or creat
SOFT_NN - Jupyter Not
localhost:8888/notebooks/SOFT_NN.ipynb
jupyter SOFT_NN Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
In [1]:
import os
import keras
Using TensorFlow backend.
In [2]:
from __future__ import absolute_import
from __future__ import division
from __future__ import print_function
import itertools
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from pylab import rcParams
import matplotlib
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
train = pd.read_csv(r'C:\Users\Nithishma\Desktop\train.csv')
print('Shape of the train data with all features:', train.shape)
train = train.select_dtypes(exclude=['object'])
print("")
print('Shape of the train data with numerical features:', train.shape)
train.drop('Id',axis = 1, inplace = True)
train.fillna(0,inplace=True)
test = pd.read_csv(r'C:\Users\Nithishma\Desktop\test.csv')
test = test.select_dtypes(exclude=['object'])
ID = test.Id
test.fillna(0,inplace=True)
test.drop('Id',axis = 1, inplace = True)
print("")
print("List of features contained our dataset:",list(train.columns))
Shape of the train data with all features: (1460, 81)
Shape of the train data with numerical features: (1460, 38)
List of features contained our dataset: ['MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces', 'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'SalePrice']
Type here to search
ENG
IN
18:14
03-11-2019


Home Page - Select or creat
SOFT_NN - Jupyter Not
localhost:8888/notebooks/SOFT_NN.ipynb
jupyter SOFT_NN Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [3]:
from sklearn.ensemble import IsolationForest
clf = IsolationForest(max_samples = 100, random_state = 42)
clf.fit(train)
y_noano = clf.predict(train)
y_noano = pd.DataFrame(y_noano, columns = ['Top'])
y_noano[y_noano['Top'] == 1].index.values
train = train.iloc[y_noano[y_noano['Top'] == 1].index.values]
train.reset_index(drop = True, inplace = True)
print("Number of Outliers:", y_noano[y_noano['Top'] == -1].shape[0])
print("Number of rows without outliers:", train.shape[0])
C:\Users\Nithishma\Anaconda3\lib\site-packages\sklearn\ensemble\iforest.py:237: FutureWarning: default contamination parameter 0.1 will change in version 0.22 to "auto". This will change the predict method behavior.
FutureWarning)
C:\Users\Nithishma\Anaconda3\lib\site-packages\sklearn\ensemble\iforest.py:247: FutureWarning: behaviour="old" is deprecated and will be removed in version 0.22. Please use behaviour="new", which makes the decision_function change to match other anomaly detection algorithm API.
FutureWarning)
Number of Outliers: 146
Number of rows without outliers: 1314
C:\Users\Nithishma\Anaconda3\lib\site-packages\sklearn\ensemble\iforest.py:415: DeprecationWarning: threshold_ attribute is deprecated in 0.20 and will be removed in 0.22.
" be removed in 0.22.", DeprecationWarning)
In [4]:
train.head(10)
Out[4]:
MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 ... WoodDeckSF OpenPorc
0 60 65.0 8450 7 5 2003 2003 196.0 706 0 ... 0
1 20 80.0 9600 6 8 1976 1976 0.0 978 0 ... 298
2 60 68.0 11250 7 5 2001 2002 162.0 486 0 ... 0
3 70 60.0 9550 7 5 1915 1970 0.0 216 0 ... 0
4 60 84.0 14260 8 5 2000 2000 350.0 655 0 ... 192
5 50 85.0 14115 5 5 1993 1995 0.0 732 0 ... 40
6 20 75.0 10084 8 5 2004 2005 186.0 1369 0 ... 255
7 50 51.0 6120 7 5 1931 1950 0.0 0 0 ... 90
Type here to search
ENG IN 18:14 03-11-2019

Home Page - Select or creat
SOFT_NN - Jupyter Not
localhost:8888/notebooks/SOFT_NN.ipynb
jupyter SOFT_NN Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [5]:
import warnings
warnings.filterwarnings('ignore')
col_train = list(train.columns)
col_train_bis = list(train.columns)
col_train_bis.remove('SalePrice')
mat_train = np.matrix(train)
mat_test = np.matrix(test)
mat_new = np.matrix(train.drop('SalePrice',axis = 1))
mat_y = np.array(train.SalePrice).reshape((1314,1))
prepro_y = MinMaxScaler()
prepro_y.fit(mat_y)
prepro = MinMaxScaler()
prepro.fit(mat_train)
prepro_test = MinMaxScaler()
prepro_test.fit(mat_new)
train = pd.DataFrame(prepro.transform(mat_train),columns = col_train)
test = pd.DataFrame(prepro_test.transform(mat_test),columns = col_train_bis)
train.head()
Out[5]:
MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 ... WoodDeckSF OpenPor
0 0.235294 0.207668 0.082802 0.625 0.428571 0.953488 0.883333 0.12250 0.416274 0.0 ... 0.000000 0.16
1 0.000000 0.255591 0.072904 0.500 0.857143 0.744186 0.433333 0.00000 0.576651 0.0 ... 0.404891 0.00
2 0.235294 0.217252 0.087396 0.625 0.428571 0.937984 0.866667 0.10125 0.286557 0.0 ... 0.000000 0.11
3 0.294118 0.191693 0.072464 0.625 0.428571 0.271318 0.333333 0.00000 0.127358 0.0 ... 0.000000 0.06
4 0.235294 0.268371 0.113835 0.750 0.428571 0.930233 0.833333 0.21875 0.386203 0.0 ... 0.260870 0.23
5 rows × 37 columns
In [6]:
COLUMNS = col_train
FEATURES = col_train_bis
Type here to search
ENG IN 18:14 03-11-2019


Home Page - Select or creat
SOFT_NN - Jupyter Not
localhost:8888/notebooks/SOFT_NN.ipynb
jupyter SOFT_NN Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [6]:
COLUMNS = col_train
FEATURES = col_train_bis
LABEL = "SalePrice"
# Columns
feature_cols = FEATURES
# Training set and Prediction set with the features to predict
training_set = train[COLUMNS]
prediction_set = train.SalePrice
# Train and Test
x_train, x_test, y_train, y_test = train_test_split(training_set[FEATURES] , prediction_set, test_size=0.33, random_state=42)
y_train = pd.DataFrame(y_train, columns = [LABEL])
training_set = pd.DataFrame(x_train, columns = FEATURES).merge(y_train, left_index = True, right_index = True)
training_set.head()
# Training for submission
training_sub = training_set[col_train]
In [7]:
y_test = pd.DataFrame(y_test, columns = [LABEL])
testing_set = pd.DataFrame(x_test, columns = FEATURES).merge(y_test, left_index = True, right_index = True)
testing_set.head()
Out[7]:
MSSubClass LotFrontage LotArea OverallQual OverallCond YearBuilt YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 ... WoodDeckSF Open
1232 0.235294 0.239617 0.074221 0.625 0.571429 0.914729 0.800000 0.00000 0.574882 0.0 ... 0.343750
76 0.352941 0.000000 0.083505 0.625 0.428571 0.891473 0.766667 0.01375 0.000000 0.0 ... 0.183043
962 0.235294 0.249201 0.091349 0.500 0.571429 0.682171 0.300000 0.00000 0.175708 0.0 ... 0.000000
433 0.294118 0.159744 0.010540 0.625 0.857143 0.271318 0.916667 0.00000 0.176297 0.0 ... 0.000000
1110 0.294118 0.306709 0.103927 0.375 0.428571 0.283566 0.000000 0.00000 0.000000 0.0 ... 0.275815
5 rows × 37 columns
In [8]:
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasRegressor
Type here to search
ENG IN 18:14 03-11-2019
Home Page - Select or creat
SOFT_NN - Jupyter Not
localhost:8888/notebooks/SOFT_NN.ipynb
jupyter SOFT_NN Last Checkpoint: 18 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [8]:
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras.wrappers.scikit_learn import KerasRegressor
seed = 7
np.random.seed(seed)
# Model
model = Sequential()
model.add(Dense(200, input_dim=36, kernel_initializer='normal', activation='relu'))
model.add(Dense(100, kernel_initializer='normal', activation='relu'))
model.add(Dense(50, kernel_initializer='normal', activation='relu'))
model.add(Dense(25, kernel_initializer='normal', activation='relu'))
model.add(Dense(1, kernel_initializer='normal'))
# Compile model
model.compile(loss='mean_squared_error', optimizer=keras.optimizers.Adadelta())
feature_cols = training_set[FEATURES]
labels = training_set[LABEL].values
model.fit(np.array(feature_cols), np.array(labels), epochs=100, batch_size=10)
880/880 [==============================] - 0s 281us/step - loss: 0.0017
Epoch 88/100
880/880 [==============================] - 0s 355us/step - loss: 0.0023
Epoch 89/100
880/880 [==============================] - 0s 306us/step - loss: 0.0018
Epoch 90/100
880/880 [==============================] - 0s 297us/step - loss: 0.0019
Epoch 91/100
880/880 [==============================] - 0s 258us/step - loss: 0.0018
Epoch 92/100
880/880 [==============================] - 0s 335us/step - loss: 0.0019
Epoch 93/100
880/880 [==============================] - 0s 328us/step - loss: 0.0017
Epoch 94/100
880/880 [==============================] - 0s 330us/step - loss: 0.0019
Epoch 95/100
880/880 [==============================] - 0s 283us/step - loss: 0.0019
Epoch 96/100
880/880 [==============================] - 0s 396us/step - loss: 0.0016
In [9]:
Evaluation on the test set created by train_test_split
Type here to search
ENG IN 18:14 03-11-2019



	A	B
1	SalePrice	Id
2	125377.46	1461
3	139463.03	1462
4	187838.8	1463
5	197409.16	1464
6	200685.45	1465
7	187992.8	1466
8	184884.81	1467
9	177838	1468
10	196438.4	1469
11	123465.02	1470
12	217517.52	1471
13	103710.68	1472
14	104635.375	1473
15	149265.44	1474
16	114954.9	1475
17	365955.9	1476
18	260562.39	1477
19	303502.47	1478
20	288211.72	1479
21	488054.6	1480
22	354869.62	1481
23	213429.3	1482
24	184882.4	1483
25	177184.34	1484